Quick guide to *Mathematica*

To get help, place the cursor on the function and press F1

To run all commands in one cell, place the cursor there and press Shift + Enter or Enter on the numerical keyboard.

Output of many commands is suppressed by ; at the end of the command. If you want to see the output, delete ;

Argument of functions must be in square brackets [...], braces {...} are for arrays, ranges etc., use double brackets [[...]] to access elements of arrays

expr /. {$x \to a$} means substitute a for x in expr, expr // function means apply function to expr, i.e. it is equivalent to function[expr].

D[f[x],x] = derivative of $f[x]$ with respect to x.

To define a function of x one can use $f[x\_] := 1 + x^2$, notice the underscore and the colon, or # nad & like in f := 1 + #² &

N[expr] evaluates expr with machine precision, if you use the decimal point in the expression it will be also evaluated with machine precision

Special characters can be inserted by pressing Esc ... Esc, e.g. Esc p Esc gives π

Clear[...] is used to unset any variables which could have been assigned previously.

Some other useful commands: Simplify, Expand, Factor; Integrate, Series, Sum

Clear all symbols from previous evaluations to avoid problems

```
In[•]:= Clear["Global`*"]
```

Fast Fourier Transform (FFT) method

Fast Fourier Transform algorithm from Numerical Recipes:

```
In[•]:= myFFT[data_, isign_] :=
  Module[
   {nn, n, m, i, j, istep, mmax, theta, tmp, wtemp, wpr, wpi, tempr, tempi, wr, wi, out},
   nn = Length[data] / 2;
   out = data;
   (* reordering using bit-reversal algorithm *)
   n = BitShiftLeft[nn, 1];
   j = 1;
   Do[
    If[j > i,
     tmp = out[[j]];
     out[[j]] = out[[i]];
     out[[i]] = tmp;
     tmp = out[[j + 1]];
     out[[j + 1]] = out[[i + 1]];
     out[[i + 1]] = tmp;
```

```
    ];
    m = nn;
    While[m ≥ 2 && j > m,
     j = j - m;
     m = BitShiftRight[m, 1];
    ];
    j = j + m,
    {i, 1, n - 1, 2}
   ];
   (* Danielson-Lanczos algorithm -
    multiplying values by factors and combining them *)
   mmax = 2;
   While[n > mmax,
    istep = BitShiftLeft[mmax, 1];
    theta = isign * 2.0 * π / mmax;
    wtemp = Sin[0.5 * theta];
    wpr = -2.0 * wtemp * wtemp;
    wpi = Sin[theta];
    wr = 1.0;
    wi = 0.0;
    Do[
     Do[
      j = i + mmax;
      tempr = wr * out[[j]] - wi * out[[j + 1]];
      tempi = wr * out[[j + 1]] + wi * out[[j]];
      out[[j]] = out[[i]] - tempr;
      out[[j + 1]] = out[[i + 1]] - tempi;
      out[[i]] = out[[i]] + tempr;
      out[[i + 1]] = out[[i + 1]] + tempi,
      {i, m, n, istep}
     ];
     (* avoiding evaluation of Sin[] in each step,
     instead trigonometric recurence is used *)
     wtemp = wr;
     wr = wr * wpr - wi * wpi + wr;
     wi = wi * wpr + wtemp * wpi + wi,
     {m, 1, mmax - 1, 2}
    ];
    mmax = istep;
   ];
   If[isign == 1,
    Return[out],
    Return[out / nn]
   ];
  ];
```

Discrete Fourier transform of lists

Mathematica built-in function (uses different normalization, $1/\sqrt{N}$ for both transformations):

```
In[ ]:= cdata = {1 + ⅈ / 2, 2 - ⅈ, 1 - ⅈ / 2, 2 + ⅈ};
        Fourier[cdata]
        InverseFourier[Fourier[cdata]]
Out[ ]= {3. + 0. ⅈ, 1. + 0.5 ⅈ, -1. + 0. ⅈ, -1. + 0.5 ⅈ}
Out[ ]= {1. + 0.5 ⅈ, 2. - 1. ⅈ, 1. - 0.5 ⅈ, 2. + 1. ⅈ}
```

Simple test of the algorithm above on this short list:

```
In[ ]:= data = Flatten[Table[{Re[cdata[[x]]], Im[cdata[[x]]]}, {x, 1, Length[cdata]}]]
        newdata = myFFT[data, 1]
        myFFT[newdata, -1]
```

Out[ ]= $\{1, \frac{1}{2}, 2, -1, 1, -\frac{1}{2}, 2, 1\}$

```
Out[ ]= {6., 0., 2., 1., -2., 0., -2., 1.}
Out[ ]= {1., 0.5, 2., -1., 1., -0.5, 2., 1.}
```

Integral Fourier transform of functions

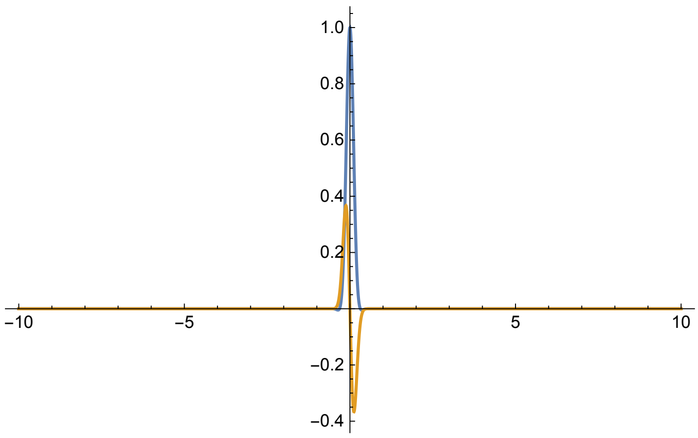
Test on the Gaussian wave packet with an example of aliasing:

```
In[ ]:= σ = 30; x0 = 0; p0 = 5;
        ψ[x_] = Exp[-σ (x - x0)^2 - ⅈ x p0]
        Plot[{Re[ψ[x]], Im[ψ[x]]}, {x, -10, 10}, PlotRange → All]
```

Out[ ]= $e^{-5\,i\,x-30\,x^2}$

... General: Exp[-2999.754862 + 49.99795714 ⅈ] is too small to represent as a normalized machine number; precision may be lost.

... General: Exp[-2999.754862 + 49.99795714 ⅈ] is too small to represent as a normalized machine number; precision may be lost.

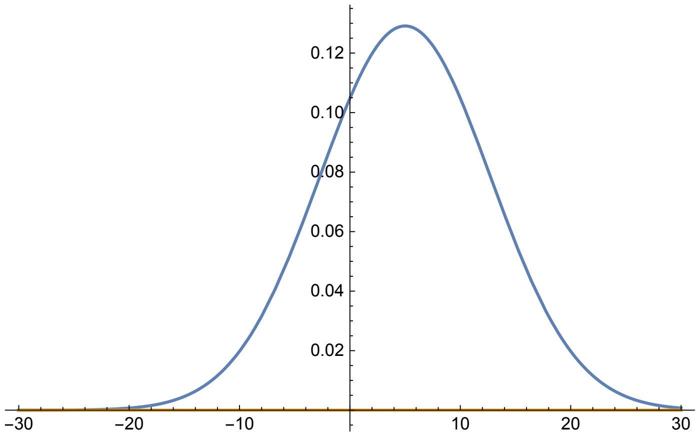
Out[ ]=


Exact Fourier transform:

In[∘]:=
```
ψp[p_] = FourierTransform[ψ[x], x, p]
Plot[{Re[ψp[x]], Im[ψp[x]]}, {x, -30, 30}, PlotRange → All]
```

Out[∘]= $$\frac{e^{-\frac{1}{120}(-5+p)^2}}{2\sqrt{15}}$$

Out[∘]=



Preparation of data to evaluate the integral Fourier transform using the fast Fourier transform (FFT):

In[∘]:=
```
n = 2^6;
Print["Number of points: ", n];
xmin = -5; xmax = 5;
h = (xmax - xmin) / (n - 1);
xpoints = Table[N[xmin + h (i - 1)], {i, 1, n}];
psix = Flatten[Table[{N[Re[ψ[xpoints[[i]]]]], N[Im[ψ[xpoints[[i]]]]]}, {i, 1, n}]];
ListPlot[{Transpose[{xpoints, psix[[1 ;; 2 * n ;; 2]]}],
  Transpose[{xpoints, psix[[2 ;; 2 * n ;; 2]]}]}, PlotRange → All]
```

Number of points: 64

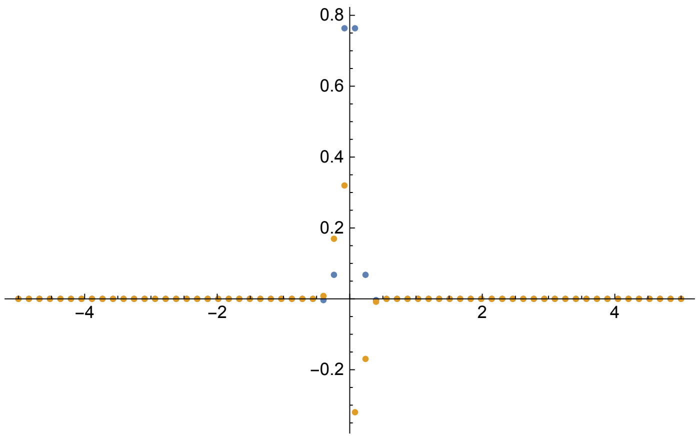
General: Exp[-750. + 25. i] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[-750. + 25. i] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[-750. - 25. i] is too small to represent as a normalized machine number; precision may be lost.

General: Further output of General::munfl will be suppressed during this calculation.

Out[∘]=



Evaluation of the integral FT using FFT and an example of aliasing:

In[∘]:=
```
Print["Nyquist frequency: ", N[π / h]];
ppoints = Table[N[2 π (-n / 2 + i - 1) / (n h)], {i, 1, n}];
psip = Flatten[Table[{N[Re[ψp[ppoints[[i]]]]], N[Im[ψp[ppoints[[i]]]]]}, {i, 1, n}]];
myFFTpsi = myFFT[psix, 1];
myCFFTpsi = Table[myFFTpsi[[2 i - 1]] + ⅈ myFFTpsi[[2 i]], {i, 1, n}];
mypsip = h / Sqrt[2 π] * Flatten[
    {Table[Exp[ⅈ 2 π (-n / 2 + i - 1) / (n h) xmin] myCFFTpsi[[n / 2 + i]], {i, 1, n / 2}],
     Table[Exp[ⅈ 2 π (i - 1) / (n h) xmin] myCFFTpsi[[i]], {i, 1, n / 2}]}];
ListPlot[{Transpose[{ppoints, psip[[1 ;; 2 * n ;; 2]]}],
   Transpose[{ppoints, psip[[2 ;; 2 * n ;; 2]]}],
   Transpose[{ppoints, Re[mypsip]}], Transpose[{ppoints, Im[mypsip]}]},
  Joined → {True, True, False, False} , PlotRange → All]
```

```
Nyquist frequency: 19.79203372
```

Out[∘]=